REMARKS

I. INTRODUCTION

In response to the Office Action dated September 23, 2004, no claims have been canceled, amended or added. Claims 1-36 remain in the application. Entry of these remarks, and re-consideration of the application, as amended, is requested.

III. PRIOR ART REJECTIONS

A. The Office Action Rejections

In paragraph (4) of the Office Action, claims 1, 3-10, 12, 13, 15-22, 24, 25, 27-34, and 36 were rejected under 35 U.S.C. §103(a) as being unpatentable over Viniotis et al., "Linear programming ... Queueing systems," IEEE, 1998 (Viniotis) in view of Schneider et al., "Stochastic Production scheduling ... demand forecasts," IEEE, 1998 (Schneider). In paragraph (5) of the Office Action, claims 2, 14, and 26 were rejected under 35 U.S.C. §103(a) as being unpatentable over Viniotis in view of Schneider and further in view of Dangat et al., U.S. Patent No. 5,971,585 (Dangat). In paragraph (6) of the Office Action, claims 11, 23, and 35 were rejected under 35 U.S.C. §103(a) as being unpatentable over Viniotis in view of Schneider and further in view of Hedlund et al., "Optimal control of hybrid systems," IEEE, 1999 (Hedlund).

Applicant's attorney respectfully traverses these rejections.

B. The Applicant's Invention

Independent claims 1, 13 and 25 are generally directed to a method for solving, in a computer, stochastic control problems of linear systems in high dimensions. Claim 1 is representative, and comprises:

(a) modeling, in the computer, a structured Markov Decision Process (MDP), wherein a state space for the MDP is a polyhedron in a Euclidean space and one or more actions that are feasible in a state of the state space are linearly constrained with respect to the state; and

(b) building, in the computer, one or more approximations from above and from below to a value function for the state using representations that facilitate the computation of approximately optimal actions at any given state by linear programming.

C. The Viniotis Reference

Viniotis describes linear programming as a technique for optimization of queueing systems. For a significant number of queueing models, that appear in diverse, seemingly unrelated application areas, such as routing, resource allocation and flow control, the optimal policy exhibits a certain "switching-curve" structure. In this paper, we formulate the optimal control problem of such models in a unified way, by using abstract Linear Programming. Using well-known facts from sensitivity analysis of Linear Programs, we show how certain properties of the optimal policy can be easily derived, even in cases where Dynamic Programming (DP) and Stochastic Dominance (SD) arguments fail. A structural property of the optimal value function of the Linear Program, namely piecewise linearity, is exploited to derive properties of the optimal cost function. We also consider additional problems in the realm of queueing system control in which DP or SD approaches are not applicable but Linear Programming may provide useful results.

D. The Schneider Reference

Schneider describes stochastic production scheduling to meet demand. Production scheduling, the problem of sequentially configuring a factory to meet forecasted demands, is a critical problem throughout the manufacturing industry. The requirements of maintaining product inventories in the face of unpredictable demand and stochastic factory output make the problem difficult. Existing approaches commonly fall into one of two groups: either demand forecasts are discarded and linearizing assumptions are made so methods based on optimal control can be applied, or AI search methods are used to tackle the large search spaces and the ability to handle stochasticity optimally is sacrificed. This paper describes a Markov Decision Process (MDP) formulation of production scheduling which captures stochasticity, while retaining the ability to construct a schedule to meet demand forecasts. The solution to this MDP is a value function, specific to the current demand forecasts, which can be used to generate optimal scheduling decisions online. The paper then describes an industrial application and a reinforcement learning method for generating an approximate value function in this domain. The results demonstrate that in both deterministic and noisy scenarios, value function approximation is an effective technique.

E. The Dangat Reference

Dangat describes a computer implemented decision support tool serves as a solver to generate a best can do (BCD) match between existing assets and demands across multiple manufacturing facilities within boundaries established by manufacturing specifications and process flows and business policies to determine which demands can be met in what time frame by microelectronics (wafer to card) or related (for example disk drives) manufacturing and establishes a set of actions or guidelines for manufacturing to incorporate into their manufacturing execution system to insure the delivery commitments are met in a timely fashion. The BCD tool has six major components, a material resource planning explode or "backwards" component, an optional STARTS evaluator component, an optional due date for receipts evaluator, an optional capacity available versus needed component, an implode "forward" or feasible plan component, and a post processing algorithm.

F. The Hedlund Reference

Hedlund describes optimal control of hybrid systems. This paper presents a method for optimal control of hybrid systems. An inequality of Bellman type is considered and every solution to this inequality gives a lower bound on the optimal value function. A discretization of this "hybrid Bellman inequality" leads to a convex optimization problem in terms of finite-dimensional linear programming. From the solution of the discretized problem, a value function that pre-serves the lower bound property can be constructed. An approximation of the optimal feedback control law is given and tried on some examples.

G. Applicant's Independent Claims Are Patentable Over The References

Applicant's claims are patentable over the references because they recite a novel and nonobvious combination of elements. None of the references, taken individually or in any combination, teaches or suggests this sequence of steps.

On page 3, the Office Action states the following:

> 4. Claims 1, 3-10, 12, 13, 15-22, 24, 25, 27-34 and 36 are rejected under 35 U.S.C. 103(a) as being unpatentable over Viniotis et al. (VI) ("Linear programming ... Queueing systems", IEEE, 1988) in view of Schneider et al. (SC) ("Stochastic Production scheduling ... demand forecasts", IEEE, 1998).

4.1 VI teaches Linear programming as a technique for optimization of queuing systems. Specifically, as per Claim 13, VI teaches solving stochastic control problems of linear systems in high dimensions (Page 652, CL1, Para 1; Page 653, CL2, Para 3); comprising:

modeling a structured Markov Decision Process (MDP) (Page 652, CL1, Para 4; Page 652, CL2 Para 6), wherein a state space for the MDP is a polyhedron in a Euclidean space (Page 654, CL2, Lemma 2);

one or more actions that are feasible in a state of the state space are linearly constrained with respect to the state (Page 653, CL1, Para 1 and Para 2; Page 652, CL2, Para 8); and

building a value function for the state using representations that facilitate the computation of approximately optimal actions at any given state by linear programming (Page 653, CL1, Para 9 to Page 654, CL1, Para 4; Page 652, CL2, Para 8).

VI does not expressly teach a computerized apparatus for solving stochastic control problems of linear systems in high dimensions comprising a computer. SC teaches a computerized apparatus for solving stochastic control problems of linear systems in high dimensions comprising a computer (Page 2726, CL1, Para 3 and 4), as that allows the solution of stochastic control problems of linear systems in high dimensions run faster and allows the user to generate the results with varying data (Page 2726, CL1, Para 3). It would have been obvious to one of ordinary skill in the art at the time of Applicant's invention to combine the method of VI with the apparatus of SC that included a computerized apparatus for solving stochastic control problems of linear systems in high dimensions comprising a computer type, as that would allow the solution of stochastic control problems of linear systems in high dimensions run faster and allow the user to generate the results with varying data.

VI does not expressly teach logic performed by the computer, for modeling a structured Markov Decision Process (MDP). SC teaches logic performed by the computer, for modeling a structured Markov Decision Process (MDP) (Page 2726, CL1, Para 3 and 4), as that allows the solution of stochastic control problems of linear systems in high dimensions run faster and allows the user to generate the results with varying data (Page 2726, CL1, Para 3). It would have been obvious to one of ordinary skill in the art at the time of Applicant's invention to combine the method of VI with the apparatus of SC that included logic performed by the computer, for modeling a structured Markov Decision Process (MDP), as that would allow the solution of stochastic control problems of linear systems in high dimensions run faster and allow the user to generate the results with varying data.

VI does not expressly teach logic performed by the computer, for building a value function for the state using representations that facilitate the computation of approximately optimal actions at any given state by linear programming. SC teaches logic performed by the computer, for building a value function for the state using representations that facilitate the computation of approximately optimal actions at any given state by linear programming (Page 2726, CL1, Para 3 and 4), as that allows the solution of stochastic control problems of linear systems

in high dimensions run faster and allows the user to generate the results with varying data (Page 2726, CL1, Para 3). It would have been obvious to one of ordinary skill in the art at the time of Applicant's invention to combine the method of VI with the apparatus of SC that included logic performed by the computer, for building a value function for the state using representations that facilitate the computation of approximately optimal actions at any given state by linear programming, as that would allow the solution of stochastic control problems of linear systems in high dimensions run faster and allow the user to generate the results with varying data.

VI does not expressly teach logic performed by the computer, for building one or more approximations from above and from below to a value function for the state using representations. SC teaches logic performed by the computer, for building one or more approximations from above and from below to a value function for the state using representations (Page 2722, CLI, Para 2; Page 2724, CL2, Para 6), as value function approximation is an effective technique for both deterministic and noisy scenarios (Page 2722, CL1, Para 2); and approximation allows solving large scale MDPs (Page 2722, CL2, Para 2). It would have been obvious to one of ordinary skill in the art at the time of Applicant's invention to combine the method of VI with the apparatus of SC that included logic performed by the computer, for building one or more approximations from above and from below to a value function for the state using representations, as value function approximation would be an effective technique for both deterministic and noisy scenarios and approximation allows solving large scale MDPs.

Moreover, on page 13, the Office Action states the following:

7.1 As per the applicant's argument that "the Office Action asserts that Viniotis teaches a state space for the MDP is a polyhedron in a Euclidean space, at Page 654, CL2, Lemma 2; however, at the indicated location, Viniotis merely states .... in Viniotis, A is a constraint matrix, not a state space; moreover, Viniotis does not refer to a polyhedron in Euclidean space", the examiner respectfully disagrees.

Viniotis states that the solution to the Linear Programming problem is an extreme point (Page 654, CL4, Para 6); extreme points form a polyhedron (Page 654, CL4, Para 6). One of ordinary skill in the art would have known that such polyhedron existed in the Euclidean space (a multi-dimensional space). The constraints of the linear program are lines in the multi-dimensional space forming the edges of the polyhedron. The constraints are defined by the states. Therefore, the state space of the linear program exists in an Euclidean space and is defined by a polyhedron. It is well known that a Markov decision Problem (MDP) is equivalent to a Linear Program; a MDP problem can be generally formulated as an equivalent Linear Program (Page 652, CL1, Para 4). Therefore, one of ordinary skill in the art would conclude that a state space for the MDP is a polyhedron in a Euclidean space.

7.2 As per the applicant's argument that "the Office Action asserts that Viniotis teaches one or more actions that are feasible in a state of the state space

are linearly constrained with respect to the state at Page 653, CL1, Para 1 and Para 2; Page 652, CL2, Para 7; however, at the indicated locations, Viniotis merely states ...it can be seen that Viniotis teaches only that a linear cost functional that involves the state is linear; however, these portions in Viniotis do not teach or suggest that actions that are feasible in a state of the state space are linearly constrained with respect to the state in the context where a state space for the MDP is a polyhedron in a Euclidian space", the examiner respectfully disagrees.

Viniotis states that the state is a linear function of the control actions (Page 652, CL2, Para 8). One of ordinary skill in the art knows that if x is a linear function of y, then y is a linear function of x. Therefore, it is clear that actions are linear functions of state. Selecting an optimal policy (set of actions) reduces to minimizing a linear functional; this minimization is constrained, since the states generated by the policy have to belong to the state space, a subset of nonnegative integers (Page 653, CL1, Para 1). Therefore, it is obvious that the actions are constrained by the state, where the state space is in the Euclidean space.

7.3 As per the applicant's argument that "the Office Action asserts that Viniotis teaches building a value function for the state using representations that facilitate the computation of approximately optimal actions at any given state by linear programming at Page 653, CL1, Para 9 to Page 654, CL1, Para 4 and Page 652, CL2, Para 8; ... Viniotis merely states ...it can be seen that Viniotis teaches only the formulation of an MDP and the definition of a value function; however, the indicated locations in Viniotis cannot be interpreted as teaching the limitations of the applicant's claim directed to "building approximations from above and from below to a value function for the state using representations that facilitate the computation of approximately optimal actions at any given state by linear programming" ", the examiner takes the position that the examiner used the above section as reference only for building a value function for the state using representations and facilitating the computation of approximately optimal actions at any given state by linear programming.

7.4 As per the applicant's argument that "the Office Action asserts that Schneider teaches building a value function for the state using representations that facilitate the computation of approximately optimal actions at any given state by linear programming at Page 2726, CL1, Para 3 and 4; ... Schneider merely states ...it can be seen that Schneider teaches only a Markov Decision Process ...", the examiner takes the position that the examiner used the above section as reference only for teaching a computerized apparatus for solving stochastic control problems of linear systems in high dimensions comprising a computer and a logic performed by a computer for modeling a structured Markov Decision Process.

7.5 As per the applicant's argument that "the Office Action states that Schneider teaches building one or more approximations from above and from below to a value function for the state using representations at Page 2722, CL1, Para 2 and Page 2724, CL2, Para 6; ... however, the indicated sections in Schneider cannot be interpreted as teaching "building approximations from above and from below to a value function for the state using representations that

facilitate the computation of approximately optimal actions at any given state by linear programming", the examiner respectfully disagrees.

Schneider teaches that the solution to the MDP is a value function and a method for generating an approximate value of this function (Page 2722, CL1, Para 2). Schneider also teaches that the solution to an MDP is an approximate value function (Page 2724, CL2, Para 6). Schneider teaches that the value function can be represented as a function of states and actions (Page 2725, CL1, Para 1). Trajectories through the MDP model are generated repeatedly using the current approximation of the value function (Page 2725, CL2, Para 4). For noisy versions, one could use noisy outcomes directly from the stochastic simulation (Page 2726, CL1, Para 3). It is inherent that when noise is introduced, the approximations to the value function will be determined by the amplitude of the noise and will thus be limited from above and from below.

Applicant's attorney disagrees. The references, taken individually or in combination, do not disclose the specific combination of elements set forth in Applicant's independent claims 1, 13 and 25.

As a general matter, the prior art simply formulates a discrete MDP in terms of linear programming, which is well known. The Applicant's invention, on the other hand, is a more general method that works in a continuous state space, continuous action setting. The Applicant's invention attempts to approximate the correct value function, with which acting optimally in each state requires solving a Linear Programming (LP) problem that incorporates this value function. The prior art does not teach or suggest these aspects of the Applicant's invention.

Turning to specifics, there are numerous examples where the references are misinterpreted by the Office Action.

For example, the Office Action asserts that Viniotis teaches "a state space for the MDP is a polyhedron in a Euclidean space," at the following locations:

<u>Viniotis: page 654, CL2, Lemma 2</u>

<u>Lemma 2:</u> If A is a totally unimodular matrix, the extreme points of the polyhedron $\{y: Ay \leq b\}$, where the vector b is integer-valued, are vectors with integer components.

<u>Viniotis: Page 654, CL1, Para 6 (NEW)</u>

Consider the LP problem (P), where now e, A, b are functions of a (vector-valued) parameter $x \in \mathbb{R}^n$. Sensitivity analysis studies how the optimal value function of (P) varies when the parameters of the model (i.e., e, A, b) vary as functions of x. In the queueing control problems of interest to us, x represents

the initial state of the queueing system. Moreover, only b depends on z, in a linear fashion. That is, $b = b_0 + F_x$, where b_0, F are (problem-dependent) constants [14].

Applicant's attorney disagrees. The Office Action imputes more into Viniotis than it actually teaches. In Viniotis, A is a constraint matrix, not a state space. Nowhere does Viniotis refer to a state space for the MDP as a polyhedron in a Euclidean space.

In another example, the Office Action asserts that Viniotis teaches "one or more actions that are feasible in a state of the state space are linearly constrained with respect to the state," at the following locations:

> Viniotis: page 653, CL1, Para 1 and 2
>
> Thus, any linear cost functional that involves the state (e.g., delay), is linear in the controls z_k. Selecting an optimal policy, therefore, reduces to minimizing a linear functional; this minimization is constrained, since the states generated by the policy have to belong to the state space S, a (possibly unbounded) subset of the nonnegative integers. From the state equation, the constraints are also linear in the control. But minimization of a linear functional over a linear constraint set is the subject of Linear Programming.
>
> There are some points that need attention. In a Linear Program, the control variables are allowed to take values in a continuum, e.g., [0,1] or $\mathbb{R}^n$. In (an unconstrained) MDP problem, the controls are integer-valued. For example, in resource allocation problems, where there are N+1 distinct actions available, $z_k \in \{0,1,\ldots,N\}$. Thus when reformulating the problem as a Linear Program, we in fact "enlarge" the solution space. This will not be a problem if existence of integer-valued optimal solutions is shown.
>
> Viniotis: page 652, CL2, Para 7
>
> In the next section we briefly present the technicalities of the formulation of the MDP problem as a linear program; we use the notation developed in [7]. The reader may find the missing details in [7,14].
>
> Viniotis: Page 652, CL2, Para 8 (NEW)
>
> Briefly, the procedure is as follows. From equation (1) (or (2)) the state is a linear function of the control actions z_k.

Applicant's attorney disagrees. The above portions of Viniotis do not teach or suggest that actions that are feasible in a state of the state space are linearly constrained with respect to the state, in the context where a state space for the MDP is a polyhedron in a Euclidean space. Instead, the above portions of Viniotis merely state that the state is a linear function of the control actions z_k.

In another example, the Office Action asserts that Viniotis teaches "building a value

function for the state using representations that facilitate the computation of approximately optimal actions at any given state by linear programming," at the following locations:

Viniotis: page 653, CL1, Para 9 to page 654, CL1, Para 4

Let Z be the set of all admissible policies; let Z_I be the subset of policies in Z that are integer-valued. Define the β-discounted, finite horizon, expected cost of policy z, when the system starts from state x at time k = 0, and is allowed to "move" for n steps (i.e., perform n transitions), as

(Eqn.(5))

where $L(z_k)$ is a linear function of the state trajectory and the control process z; it has the interpretation of an instantaneous cost. A fairly general form for L, that fits our purposes is

(Eqn.(6)

where c, d are properly dimensioned vector constants. In resource allocation problems, where delay is the cost, we have d = 0; in pure blocking systems, we choose c = 0.

To show the exact dependence of J,,(z, z) on x and z, let us rewrite (4) as

(Eqn.(7))

Then since x is constant and (Eqn.), where p denotes the probability distribution on Ω^n, we have

(Eqn.(8))

Equation (8) stresses the fact that the cost function is linear in the variables $z_k(w^k)$.* The dependence of the cost on the probability distribution, the transitions and the constants c,d is "hidden" in $\gamma_k(w^k)$, to emphasize the dependence of the cost on the policy z. The exact form of $\gamma_k(w^k)$ can be routinely determined for the specific problem in hand [14]. We need only mention that $\gamma_k(w^k)$ is independent from the control policy z and the initial state x. For the purposes of the discussion in this section, the exact form of $\gamma_k(w^k)$ is irrelevant.

From (8) we see that the optimal policy is the one that minimizes the second term in the right hand side. From (7) the constraints fall in general into two categories:

(a) nonnegativity of states, namely

(Eqn.(9))

(b) boundedness of states, namely

(Eqn.(10))

where U is the bound. Since the constraints in (10) (≤) are easily converted into constraints as in (9), we shall concentrate on constraints of the form (9) only.

Summarizing, the LP equivalent problem may take the form

min eZ (P)

AZ ≤ b

This form is suitable to present results from sensitivity analysis.

Remark. The control variables are (Eqn.), and thus there is only a finite number of them. The constraint matrix A has elements that depend only on the transitions $\xi_k(w^k)$. The vector b depends only on the initial state z.

We have allowed $z_k(w^k)$ to take values in [0,1]. For sensitivity analysis, x, the initial state of the queueing system, should be also continuously-valued. In this case, the trajectory i will be continuously-valued; such a trajectory does not of course correspond to a real queueing system.

If, however, $x, z_k(w^k)$ are restricted to take integer-valued values only, then i will be integer-valued; in this case it does represent the evolution of the queueing system. The optimal cost function of the MDP in this case is given by*

(Eqn.(11))

This is actually a problem in Integer Programming, the sensitivity analysis of which is not as well developed as that of a Linear Program. If we remove the restriction on integer-valued policies (and states), we have the above mentioned Linear Programming problem (P). Let

(Eqn.(12))

denote the optimal value function of problem (P). It is $W_n(z)$ for which results from sensitivity analysis apply. We wish to emphasize here that the functions W_n, V_n are quite different; first of all, they are even defined on different domains. If we can make, however, a suitable connection between them, then we can relate the properties of W_n (which we shall determine) to those of V_n (which we want).

Such a connection is indeed possible, if the Linear Program in (12) admits an integer-valued solution. In this case, for integer-valued x, (11) and (12) refer to the same problem. The optimal value function of the LP "contains" in some sense the optimal value of the MDP: we can recover $V_n(x)$ by "interpolating" $W_n(x)$ at the integer-valued points of its domain. Consequently, all the properties of $W_n(x)$ are automatically properties of $V_n(x)$ as well.

<u>Viniotis: page 652, CL2, Para 8</u>

Briefly, the procedure is as follows. From equation (1) (or (2)) the state is a linear function of the control actions z_k.

Applicant's attorney disagrees. The above portions of Viniotis do not teach that building a value function for the state using representations and facilitating the computation of approximately optimal actions at any given state by linear programming. Instead, the above portions of Viniotis teach only the formulation of an MDP and the definition of a value function, as well as that the state is a linear function of the control actions.

In another example, the Office Action asserts that Schneider teaches "building one or more approximations from above and from below to a value function for the state using representations that facilitate the computation of approximately optimal actions at any given state by linear programming," at the following locations:

Schneider: page 2726, CL1, Para 3 and 4

Our experiments consider both deterministic and noisy versions of the problem. To build the deterministic version of the problem, we ran long (stochastic) simulations for each of the 421 actions and cached the mean observed production rate for each. For the noisy versions, we could have used noisy outcomes directly from the stochastic simulation, but instead we simply added Gaussian noise to the cached, deterministic production rates. This enabled our experiments to run significantly faster, and also allowed us to easily generate empirical results with varying amounts of noise.

Table 1 shows experimental results. The computation times reported are on a 200 MHz Pentium Pro. The first section contains results for the case where the factory output is deterministic and known. The purpose of the first two lines is to delimit the range of results we should expect from good algorithms. The "Random" algorithm builds a schedule by choosing 8 configurations at random, and it loses an enormous amount of money. Much of the cost is due to heuristic penalties for failing to satisfy customer demand.

Schneider: page 2722, CL1, Para 2

In this paper, we describe a Markov Decision Process (MDP) formulation of production scheduling which captures stochasticity, while retaining the ability to construct a schedule to meet demand forecasts. The solution to this MDP is a value function, specific to the current demand forecasts, which can be used to generate optimal scheduling decisions online. We then describe an industrial application and a reinforcement learning method for generating an approximate value function in this domain. Our results demonstrate that in both deterministic and noisy scenarios, value function approximation is an effective technique.

Schneider: page 2724, CL2, Para 6

Here we describe a principled approach to generating closed-loop production scheduling policies with reinforcement learning methods. It combines the capabilities of both optimal control and AI search based methods. The approach is based on representing the problem as an MDP and representing the solution as an approximate value function. In contrast to many optimal control based methods, it produces a time-dependent policy specifically built to match current demand forecasts, rather than a time-invariant policy that ignores all demand information other than the current rate. Our experiments also demonstrate the ability to search hundreds of alternative factory configurations.

Schneider: Page 2725, CL1, Para 1 (NEW)

Abstractly, a Markov Decision Process (MDP) is defined by a state space X, action set A, immediate reward function R(x, a), and probabilistic transition model P(x'|x, a). The solution to the MDP is a policy $\pi^*: X \rightarrow A$ which, if followed by the agent, will maximize the expected long-term sum of rewards attainable starting from any state x. Dynamic programming methods tabulate this optimal cumulative reward in the optimal value function $V^*(x)$, which is the unique solution to the Bellman equations [3]:

(Eqn. 1)

Once V* is computed, the optimal policy π^* is immediately obtained by choosing any action which instantiates the max in Eq. 1.

Schneider: Page 2725, CL2, Para 4 (NEW)

• The action set consists of all legal factory configurations. We assume a discrete-time model, so the configuration chosen at time t will run unchanged until time t + 1.

Applicant's attorney disagrees. The above portions of Schneider do not teach or suggest building one or more approximations from above and from below to a value function for the state using representations that facilitate the computation of approximately optimal actions at any given state by linear programming. Instead, the above portions of Schneider teach only a Markov Decision Process (MDP) formulation of production scheduling which captures stochasticity. Further, the above portions of Schneider merely describe how a Markov Decision Process (MDP) is defined by a state space X, action set A, immediate reward function R(x, a), and probabilistic transition model P(x'|x, a). Finally, the above portions of Schneider merely describe how the solution to the MDP is an approximate value function, specific to the current demand forecasts, which can be used to generate optimal scheduling decisions online.

Dangat and Hedlund fail to overcome these deficiencies in the combination of Viniotis and Schneider. Recall that Dangat and Hedlund were cited only against the dependent claims.

The various elements of Applicant's claimed invention together provide operational advantages over Viniotis, Schneider, Dangat, and Hedlund. In addition, Applicant's invention solves problems not recognized by Viniotis, Schneider, Dangat, or Hedlund.

Thus, Applicant submits that independent claims 1, 13, and 25 are allowable over Viniotis, Schneider, Dangat, and Hedlund. Applicant's dependent claims 2-12, 14-24, and 26-36 are submitted to be allowable over Viniotis, Schneider, Dangat, and Hedlund in the same manner, because they are dependent on independent claims 1, 13, and 25, respectively, and thus contain all the limitations of the independent claims. In addition, dependent claims 2-12, 14-24, and 26-36 recite additional novel elements not shown by Viniotis, Schneider, Dangat, or Hedlund.

IV. CONCLUSION

In view of the above, it is submitted that this application is now in good order for allowance and such allowance is respectfully solicited. Should the Examiner believe minor matters still remain that can be resolved in a telephone interview, the Examiner is urged to call Applicant's undersigned attorney.

Respectfully submitted,

GATES & COOPER LLP
Attorneys for Applicants

Howard Hughes Center
6701 Center Drive West, Suite 1050
Los Angeles, California 90045
(310) 641-8797

Date: November 23, 2004

By:
Name: George H. Gates
Reg. No.: 33,500

GHG/